5-1-2015

THE EFFECT OF THE ANCILLA VERIFICATION ON THE QUANTUM ERROR CORRECTION

Ali Abu-Nada
Southern Illinois University Carbondale, anada@siu.edu

THE EFFECT OF THE ANCILLA VERIFICATION ON THE QUANTUM ERROR CORRECTION

By

Ali Abu-Nada
M.S. Southern Illinois University at Carbondale

A Dissertation
Submitted in Partial Fulfillment of the Requirements for the
Doctor of Philosophy Degree

Department of Physics
in the Graduate School
Southern Illinois University Carbondale
May, 2015

DISSERTATION APPROVAL

THE EFFECT OF THE ANCILLA VERIFICATION ON THE QUANTUM ERROR CORRECTION

By

Ali Abu-Nada

A Dissertation Submitted in Partial

Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in the field of Physics

Approved by:

Mark Byrd, Chair

Eric Chitamber

Saikat Talapatra

Thushari Jayasekera

Jerzy Kocik

Graduate School

Southern Illinois University Carbondale

February, 13, 2015

AN ABSTRACT OF THE DISSERTATION OF

ALI ABU-NADA, for the Doctor of Philosophy degree in APPLIED PHYSICS, presented on 02/13/2015, at Southern Illinois University Carbondale

TITLE: THE EFFECT OF THE ANCILLA VERIFICATION ON THE QUANTUM ERROR CORRECTION

MAJOR PROFESSOR: Dr. Mark Byrd

Communication is the prototypical application of error-correction methods. To communicate, a sender needs to convey information to a receiver over a noisy "communication channel." Such a channel can be thought of as a means of transmitting an information-carrying physical system from one place to another. During transmission, the physical system is subject to disturbances (noise) that can adversely affect the information carried. To use a communication channel, the sender needs to encode the information to be transmitted in the physical system. After transmission, the receiver decodes the information. Quantum error correction is used in quantum computing to protect quantum information from errors due to decoherence and other quantum noise. Quantum error correction is essential if one is to achieve fault-tolerant quantum computation that can deal with both noise on stored quantum information, and also with faulty quantum gates, faulty quantum preparation,and faulty measurements.

In this dissertation, we look at how additional information about the structure of the quantum circuit and noise can improve or alter the performance of techniques in quantum error correction.

Chapter 1 and 2, are an introduction to the quantum computation, quantum error correction codes and fault-tolerant quantum computing. These chapters are written to be a useful for students at the graduate and advanced undergraduate level. Also. The first two chapters of this dissertation will be useful to researchers in other fields

who would like to understand how quantum error correction and fault-tolerant quantum computing are possible.

In chapter 3, we present numerical simulation results comparing the logical error rates for the fault-tolerant [[7, 1, 3]] ’s 7 code using the technique of ancilla verification vs. the newer method of ancilla decoding as described in [1].

In chapter 4, we determine how often one should apply error correction. Therefore, we provide a relationship between the logical error rate and the physical error rate for a sequence of logical gates, sometimes followed by noisy quantum error correction.

DEDICATION

To:
The sake of my GOD (ALLAH), my Creator and my Master...

To:
My great teacher and messenger, Mohammed (May ALLAH bless and grant him), who taught us the purpose of life...

To:
My great late parents, brothers and sisters, my wife, and my sons...

To:
The soul of Professor Bary Malik...

ACKNOWLEDGMENTS

I would like to gratefully and sincerely thank Professor Byrd for his guidance, understanding, patience, and most importantly, his friendship during my graduate studies at SIUC. His mentorship was paramount in providing a well rounded experience consistent my long-term career goals. He encouraged me to not only grow as a researcher or physicist but also as an instructor and an independent thinker. I am not sure many graduate students are given the opportunity to develop their own individuality and self-sufficiency by being allowed to work with such independence. For everything you have done for me, Professor Byrd, I thank you.

I was very Lucky to meet Dr.Ben Fortescue and Professor Eric Chitamber, I had the pleasure of hearing Dr.Fortescue talks on fault-tolerant error correction, and Prof. Chitamber talks on open quantum system. This project will not be accomplished without a great help from Dr. Fortescue.

I would like to thank my committe members: Prof. Eric Chitamber, Prof. Saikat Talapatra , Prof. Thushari Jayasekera , and Prof. Jerzy Kocik for taking some time of their busy schedules to read and hear about my work. This dissertation would not have been possible without the help and advice of Dr. Naushad Ali and Dr. Saikat Talapatra. I would like to thank the Department of Physics staff: Sally Pleasure, Suzanne McCann, and Bob Baer, they were always very kind and helpful.

I would like to thank my friends Kamal Naser and Yaser Naser from outside my SIUC circle who supported me with their kind words, thoughts, prayers and blessings.

In the end none of this would have been possible without the unconditional and unwavering love of my wife and my sons (Mahmoud and Amir), and the memory of my days as a child growing up in Amman.

TABLE OF CONTENTS

LIST OF TABLES

TABLE PAGE

LIST OF FIGURES

FIGURE PAGE

CHAPTER 1

BACKGROUND

This chapter aims to give a brief introduction to the terminology and techniques of quantum computing, error correction and fault tolerance, for full review see [2–4].

1.1 Overview: Quantum computer vs. Classical Computer

A quantum computer is a computation device that makes direct use of quantum mechanical phenomena, such as superposition and entanglement, to perform operations on data. Quantum computers are different from digital (classical) computers based on transistors. Whereas, digital computers require data to be encoded into binary digits (bits) where each bit represents either a one or a zero, quantum computation uses quantum properties to represent data and perform operations on these data [2]. In quantum computing, experiments have been carried out in which quantum computational operations were executed on a very small number of quantum bits note as qubits. A single qubit is represented by two states of a quantum mechanical system [5], which are represented by $|0\rangle$and $|1\rangle$. Moreover, a pair of qubits can be in any quantum superposition of 4 states, and three qubits in any superposition of 8. In general, a quantum computer with n qubits can be in an arbitrary superposition of up to 2^n different states simultaneously (this compares to a normal computer that can only be in one of these 2^n states at any one time). Large-scale quantum computers will be able to solve certain problems much faster than any digital computer using the best currently known algorithms.
The main and general question for now is, why do we want to build a quantum computing device rather than classical computing device? The answers of this question are the followings [6]:

1. To make computers faster and more compact, we have been making them smaller. However, there is a limit to how much smaller we can make them and

still have them function as they do now. This is due to quantum mechanics. In other words, the limit to small scale computational technology is governed by quantum mechanics, since, at a certain scale, the current computational systems will not be able to be approximated by Newtonian mechanics. So, to make things smaller, we need to use quantum mechanics.

2. We know how to solve some problems much more efficiently on a quantum computer than any classical one. These include factoring large integers (used in present-day cryptography), searching an unsorted database (done regularly by many people), and simulating quantum systems. Simulating quantum systems more efficiently could lead to better materials, better drugs, and cleaner energy extraction methods. This is because all of these problems involve many-body quantum systems.

3. Quantum information can be used in a variety of ways beyond computing. Such as quantum cryptography, quantum games, and quantum communication of all sorts. It therefore has the potential to revolutionize our society since we live in the so-called "information age."

1.2 Qubit States

As mentioned above, a qubit could be in one of two possible states $|0\rangle$ and $|1\rangle$, which correspond to the states 0 and 1 for classical data bit. Dirac's bracket notation is used herein to denote quantum states. The difference between bits and qubits is that a qubits could also be in a state $|0\rangle$ and $|1\rangle$ as well as in a linear combinations of these two, often called superpostion [6]. It is this superposition state which makes the qubit very different from a classical bit. There is no classical analogue for the quantum mechanical superposition states, which can be more highly correlated than any classical state. One such example is the state

$$\psi = \frac{1}{\sqrt{2}} (|000\rangle + |111\rangle) \tag{1.1}$$

A measurement on the first of the states in the superposition will imply that the other states are the same state as the first. However, each is obtained with a

probability of one-half. We can think of this as a massively parallel operation since the number of zeroes and ones can be very large. (We could have n rather than four, where n could be a million or more.)

1.3 Quantum computation

A computation is a process which maps bit strings to other bit strings, either in a deterministic way, or with some probability. We can decompose any classical computation into a sequence of fundamental operations, called gates, and a connectivity diagram, a wiring, indicating how the outputs of one gate are mapped into the inputs of future gates. In section 1.2 we studied that a qubit would be described by a vector

$$|\psi\rangle = \alpha\,|0\rangle + \beta|1\rangle \tag{1.2}$$

where the components α and β are both complex numbers, called amplitudes.
We can perform a measurement on our qubit, which returns a classical bit, 0 or 1. The ”logical basis“ or ” computational basis“ measurement returns the value 0 with probability $|\alpha|^2$ and the value 1 with probability $|\beta|^2$. After the measurement, if we received the result 0, then we will find our qubit in the state$|0\rangle$, and if we received the result 1, then we will find our qubit in the state $|1\rangle$. In effect, the measurement projects the state of the qubit onto the two basis states, $|0\rangle$and $|1\rangle$. The normalization condition means that we will always find our qubit in one of the two states.
We can also perform a measurement in a different basis. One important basis is the “dual basis” measurement, which projects onto the two states.

$$|+\rangle = \frac{1}{\sqrt{2}}\,(|0\rangle + |1\rangle) \tag{1.3}$$

$$|-\rangle = \frac{1}{\sqrt{2}}\,(|0\rangle - |1\rangle) \tag{1.4}$$

where we will receive the first outcome with probability $|\alpha + \beta|^2$, leaving the qubit in the state $|+\rangle$, and the second with probability $|\alpha - \beta|^2$, leaving the qubit in the state $|-\rangle$. In general, we can project onto any orthogonal set of basis states.

The state of the qubit can be reversibly manipulated by applying a unitary operation - a linear map from one state $|\psi\rangle$ to a new state $|\psi'\rangle$. For a single qubit, such a map can be thought of as a 2 $\times$ 2 complex matrix. The linear operator that satisfies the condition of matrix representation $U^{\dagger}U = I,$ where I is the 2 $\times$ 2 identity matrix ,and $U^{\dagger}$ is the adjoint of U . Examples of a single qubit operations are Pauli operations. They are a set of 2 $\times$ 2 unitary and Hermitian matrices. In this section we describe some simple quantum operations (gates).

1.3.1 The Pauli operators

Quantum computer circuits consists of wires and logic gates . The wires are used to carry information around the circuits, while the logic gates perform manipulations of the information, converting it from one from to another. An example of these single qubit gates are Pauli gates (operations).

$$I \equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},\ X \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, Z \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, Y \equiv \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad (1.5)$$

We can describe the computational basis measurement as a measurement of the eigenvalues of Z, and the dual basis measurement corresponds to measuring the eigenvalues of X; we could similarly measure the eigenvalues of the Y operation. For a qubit consisting of a particle's spin, these three measurements correspond to measuring the spin along the x, y and z axes. Each Pauli operation describes a useful basis, defined by its eigenvectors [7].

1.4.2 The Clifford group

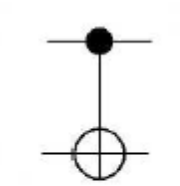
Another useful group of operations are the Clifford group of operations C_1 , which are defined to be the operations which transform Pauli operations into Pauli operations $CPC^{\dagger} \in \mathrm{P_n}$ for P $\in \mathrm{P_n}$, C $\in C_1$. Examples of Clifford operations include the Hadamard operation

$$H \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (1.6)$$

which exchanges the logical and dual bases; the phase gate

$$S \equiv \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix} \tag{1.7}$$

and the two-qubit controlled-NOT or CNOT gate



$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

which in the logical basis flips the value of the second qubit based on the value of the first. In fact, any operation in the Clifford group can be decomposed into these elementary operations [7].

CHAPTER 2

FAULT TOLERANT QUANTUM ERROR CORRECTION CODES

2.1 Background: the need for error correction

Quantum computers have a great deal of potential, but to realize that potential, they need some sort of protection from noise. Classical computers don't use error correction. One reason for this is that classical computers use a large number of electrons, so when one goes wrong, it is not too serious. A single qubit in a quantum computer will probably be just one, or a small number, of particles, which already creates a need for some sort of error correction. Another reason is that classical computers are digital: after each step, they correct themselves to the closer of 0 or 1. Quantum computers have a continuum of states, so it would seem, at first glance, that they cannot do this. For instance, a likely source of error is over-rotation: a state $\alpha\,|0\rangle + \beta\, e^{-i\varphi}|1\rangle$ might be supposed to become $\alpha\,|0\rangle + \beta\, e^{-i(\varphi+\delta)}|1\rangle$.

The actual state is very close to the correct state, but it is still wrong. If we do not do something about this, the small errors will build up over the course of the computation, and eventually will become a big error [10].

2.2 Classical error correction: the Repetition Code

The next example is a special case of the main problem of classical error-correction and occurs in typical communication settings and in computer memories. Let the physical system consist of three bits. The effect of the errors is to independently flip each bit with probability p, which we take to be p = 0.25.

Table 2.1: Error syndromes for a three bit repetition cods

Error syndrome	Code word	Error's location
00	000 or 111	No error
01	001 or 110	Bit 3 flipped
10	100 or 011	Bit 1 flipped
11	010 or 101	Bit 2 flipped

The repetition code, results from triplicating the information to be protected. An encoding is given by the map $0 \rightarrow 000$,$1 \rightarrow 111$, These are called an encoded zero/one state or a logical zero / one state. The repetition code is the set {000, 111}, which is the range of the encoding. The information can be decoded with majority logic: If the majority of the three bits is 0, output 0, otherwise output 1 [11].

How well does this encoding/decoding combination work for protecting one bit of information against the errors? The decoding fails to extract the bit of information correctly if two or three of the bits were flipped by the error. If an error occurs, we get one of the strings 001, 010,100, 110, 101, 011.

This is the simplest possible error-correcting code, the majority-rule code. We can frame the majority rule code in terms of parities as well. In this case, we look at two parities: the parity of the first two bits, and the parity of the second two bits.

For the legitimate code words 000 and 111, these both have parity 0; for all other strings, at least one of them has parity 1. We call these values parity checks or error syndroms. Thus, if we know the syndrome, we can correct the error by flipping the corrupted bit again.

2.3 Quantum Error Correcting Code

Error correction in general is accomplished by redundantly encoding information. The simplest classical example is duplicating the information several times and recover any error by doing a

majority voting. However, in the quantum realm this approach is unfeasible. The first obstacle is the no-cloning theorem [12], which states that in the quantum world it is not possible to make a perfect copy of an arbitrary unknown quantum state. In another word,there is no universal quantum copying machine that will take in a state and put out two copies of the same state for any given input state. Secondly, a straightforward evaluation of the majority voting is not possible, because a direct measurement would destroy any coherent quantum superposition. Finally, classical error correction has to deal with a single type of error only, namely a flip of a bit, whereas quantum states can suffer from a continuum of possible errors, that can even add up over time. Shor [13] and Steane [14] discovered a way to overcome these objections. The key idea is to encode a quantum state in a highly entangled state of additional supporting qubits. Thus, a small subspace of the system's Hilbert space is defined as the code subspace. This is chosen such that possible errors move the code subspace to mutually orthogonal error subspaces of the system. To avoid collapse of the quantum superposition by measurement operations it is necessary to extract the error information, that indicates a potential error, by partial measurement. The measurement result is called the error syndrome and gives information about the error only, without revealing information about the data itself. Linear combinations of correctable error are also correctable in the sense that the syndrome measurement projects the state into a well defined error subspace which can then be corrected by applying the appropriate unitary transformation which reverses the effect of the error. It is fundamental to quantum error correction that the ability to correct a discrete set of errors suffices to correct a much larger, even continuous class of errors [15].

Over time many different quantum error correction codes (QECC) had been developed. They are classified as [[n, k, d]] QECCs where k logical qubits are encoded in n physical qubits protecting against errors of distance d. The Hamming distance d comes from classical coding theory and states that going from any codeword in the code to any other codeword requires a flip of at least d bits. A linear code with distance of at least (2t + 1) can correct errors on up to t bits [2].

We will concentrate on codes with k = 1 and distance d = 3, which can correct an arbitrary error on a single logical qubit. An analysis of the performance of higher distance codes can be

found in [16]. The first QECC that we describe is Shor's 9-qubit code [13], which first showed a way out of the conundrum of effective quantum error correction. Even more important is Steane's 7-qubit.

2.3.1 Bit-flip Errors: A Quantum Code

The quantum bit-flip code uses three quantum states to encode one as does the classical bit-flip code above. The state $|0\rangle\otimes |0\rangle\otimes |0\rangle= |000\rangle= |0\rangle_L$ is the logical state representing the zero state of the encoded qubit. (The subscript L is to indicate that it is a logical state and the b indicates that it is a bit-flip code. We will see below why this distinction is helpful.) Similarly, $|1\rangle\otimes |1\rangle\otimes |1\rangle= |111\rangle= |1\rangle_L$ is used for the logical one state.

Note that one cannot just clone a state to produce redundancy due to the No-Cloning Theorem. Also, the encoded state needs to preserve superpositions such as

$$|\psi\rangle = \alpha |0\rangle + \beta|1\rangle \tag{2.1}$$

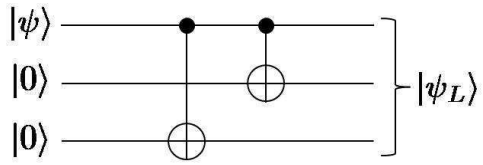


Figure 2.1: Circuit diagram for encoding a qubit into a 3-qubit bit-flip protected code

To encode the state redundantly, cloning is not required. The encoding can be accomplished using the CNOT gate twice. Simply apply $CNOT_{13}$ and $CNOT_{12}$ to the following state of three qubits,

$$|\psi\rangle|00\rangle = \alpha |0\rangle + \beta|1\rangle|00\rangle$$

This will produce :

$$|\psi_L\rangle = \alpha |000\rangle + \beta|111\rangle \tag{2.2}$$

Error syndrome extraction:

Now a method for measurement and recovery is needed. The problem is that in quantum

mechanics one cannot just measure the three states to see if they agree; a quantum state can be in a superposition of the (logical) zero state and the (logical) one state as above, and a measurement of the first qubit to see if it is in the state zero or not will immediately produce the state $|000\rangle$with probability $|\alpha|^2$ and $|111\rangle$with probability $|\beta|^2$ thus destroying the superposition of the qubit state. The state would then contain only classical information. (Essentially it is equivalent to the classical 000 or 111 binary state.) Since we need to preserve arbitrary superpositions, we cannot use this method for determining whether or not an error occurred. Now let us suppose that a bit-flip error occurs on $|\psi_L\rangle$. The objective is to determine if the state has experienced a bit-flip error or not without ruining the superposition and, if it has an error, to determine which qubit experienced the error. This can be done by checking to see if the first two qubits are the same or not and then checking to see if the last two qubits are the same or not without ever determining whether the state is the logical zero, logical one, or a superposition of the two. Let us examine this process in detail. First, notice the state |0i is an eigenvector of Z with eigenvalue 1 and $|1\rangle$is an eigenvector of Z with eigenvalue −1. Then any logical state is an eigenstate of the operator $Z \otimes Z \otimes I$ with eigenvalue of 1 if the first two qubits are the same and −1 if they differ. For example,

$$(Z \otimes Z \otimes I)\,|\psi_L\rangle = (Z \otimes Z \otimes I)\alpha\,|000\rangle + \beta|111\rangle \tag{2.3}$$

Of course the same is also true for the operator $Z \otimes Z \otimes I$. However, suppose that a bit-flip error occurs on the first qubit, giving

$$(X \otimes \mathrm{I} \otimes I)\,|\psi_L\rangle = (X \otimes I \otimes I)\alpha\,|000\rangle + \beta|111\rangle = \alpha\,|100\rangle + \beta|011\rangle \tag{2.4}$$

Then, If we measure $Z \otimes Z \otimes I$ on the new state, we will get -1.

However, it does seem that the error can be detected. Since determining the value of the operator $Z \otimes Z \otimes I$ shows that the last two qubits agree, we know that the error occurred on the first qubit. In fact, it is not difficult to convince yourself that measuring these two operators will determine which qubit experienced a bit-flip for any of the three. Just like the classical bit-

flip code, this will not indicate whether or not an error occurred on two qubits. Thus the probability must be small, just like the case for the classical code. We call $Z \otimes Z \otimes I$ and $I \otimes Z \otimes Z$ are the stabilizers of the state ($\alpha\,|000\rangle + \beta|111\rangle$), since measuring these operators will result an +1 eigenvalue. Now, we have the idea that we could determine the parity of the pairs of qubits to determine if they are the same or different. But how would we determine this in practice? A method for doing this is shown in Figure 2.2 gives a circuit for determining the error, also known as a syndrome measurement . In this example, a bit-flip error occurred on qubit 1 in the 3-qubit QECC. This is represented by an X gate. After 4 CNOT gates, the two ancillary qubits are measured. A measurement in the $|0\rangle$, $|1\rangle$basis gives a result of $|1\rangle$ for the top ancillary qubit and $|0\rangle$for the bottom one. This tells us that the first qubit has had a bit-flip error. We then feed this information back into the system by implementing an X gate on the first qubit, thus correcting the error. Notice that we have not determined the coefficients of the superposition of the logical zero and logical one states. We have only determined that there was an error on the first qubit since it does not agree with the other two. (Assuming that only one bit-flip error could have occurred.)

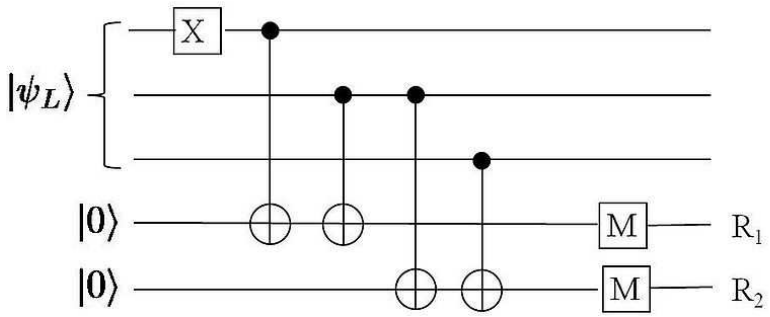


Figure 2.2: A method for extracting a bit-flip error syndrome from a 3-qubit bit-flip protected code. The M's are measurements on the ancillary qubits, the results of which are recorded as R_1 and R_2 measuring

Phase-flip Errors:

”Phase-flip errors“ are errors which change the sign of the |1i state. This is not a classical error as it does not occur on a classical bit. However, it does occur on qubits that are not in the zero state. Thus these errors must be treated. Much of what works for the bit-flip errors also works for phase-flip errors once we are able to encode properly. Let us consider the following states that we will used to encode our logical qubit:

$$|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle) \quad (2.5)$$

In this case, when a ”phase-flip“ occurs, the |+i becomes a |−i or vice versa. Therefore it is similar to the bit-flip error since there are two orthogonal states that are changed into one another by the error. In this case the error operator is of the form Z. As before, if a phase error occurs on the first qubit, then we can encode redundantly by letting $|0\rangle_L = |+++\rangle$ and $|1\rangle_L = |---\rangle$ It is easy to see that this code will enable the detection and correction of one phase

error just as the bit-flip code did for one bit-flip. In this case we exchange the Z in the bit-flip code with a X for the phase-flip code and the process carries through as before.

2.4 Fault Tolerance

As the name implies, fault-tolerance means that quantum computations can be performed in spite of errors in the computation. To ensure that a computation is reliable, one must be able to prevent errors from accumulating. This could happen, for example, if a small error occurs on one qubit and propagates to many others before it is fixed. What are all the ways in which an error can occur and how can they be prevented from accumulating to produce erroneous results? In this chapter, these questions are addressed [6].

2.4.1 Requirements for Fault-Tolerance

Preskill give five laws for reliable quantum computing [17], as he reviews the results obtained for avoiding failure. Here, a slightly modified list is discussed. The list is:

- Don not use the same ancilla twice,
- Copy errors not data,
- Carefully prepare ancilla,
- Verify ancilla,
- Verify the syndrome,

- Take care with measurements.

All of these require some explanation. Let us take them in order.

Don not use the same qubit twice:

For example, one should be careful when qubits are reused because error correction procedures can actually propagate errors. Consider the syndrome measurement in Figure 2.2. In that circuit, one of the ancillary qubits is used twice to check the parity of a pair of qubits in the bit-flip code.

CHAPTER 3

RELATIVE PERFORMANCE OF ANCILLA VERIFICATION AND DECODING IN THE [[7,1,3]] STEANE CODE

3.1 Introduction

3.1.1 Ancilla post-selection in quantum error correction

Quantum error-correcting codes (QECCs) provide a means to protect quantum data against noise by encoding quantum states into larger Hilbert spaces such that some class of error operations are correctable [13, 14]. In a physically-realistic system, one must take into account that the correction operations themselves must be implemented using imperfect quantum operations. In order for quantum error correction (QEC) operations to be effective in protecting the data, they (and other operations on the data) must be implemented in a fault-tolerant way [4], i.e., in a way such that a single faulty quantum gate cannot lead to multiple errors on the data (section 2.7). Some logical operations are naturally fault-tolerant, such as transversal gates in which every physical qubit is acted on by a separate gate, and the absence of any “cross-talk" between different physical qubits in an encoded block means there is no opportunity for errors to spread within that block. However, QEC generally involves ancillary states which are used to carry away entropy and purge the data of errors. These states must be very carefully prepared so that they do not spread errors to the data and this preparation must also operate fault-tolerantly. Often, however, non-fault-tolerant circuits must be used for the initial ancilla preparation, and fault-tolerance is instead enforced by post-selection (ancilla verification) in which only those ancillas which satisfy some measurement outcome after being created are subsequently used to interact with the data [13, 14]. Thus, it is not known beforehand whether a given ancilla will be used, and one may need to perform multiple ancilla creations before obtaining one which satisfies the post-selection criterion.

This raises the question of how to ensure that a post-selected ancilla is available forQEC with sufficiently high probability so as not to significantly increase the overall failure rate of the QEC. One may suppose that if an ancilla fails to pass verification, then the data would wait until a suitable ancilla is verified. However, this is quite impractical since at any given point in

the circuit, a large computation may have many parallel gating operations simultaneously performed. If the data is required to wait, many qubits may have to wait, which could lead to many errors. Furthermore, one must ideally have these ancillas created “as close as physically possible” to the data in order to avoid additional movement operations and associated errors. Two obvious approaches are to either: 1) create a limited number of ancillas sequentially until post-selection is passed and to time the preparation so that it is likely that one will be available for use when needed, or 2) to create several ancillas in parallel so there is a high probability that at least one will pass [19]. Both approaches have somewhat analogous disadvantages. In the first case, if an ancilla passes in the first try, sequential creation requires the ancilla to wait (and, in general, accumulate errors) until the ancilla will be used. Parallel creation avoids this, but has the problem that in a given physical architecture there will be a very limited number of ancillas that can be created close to the data. Thus, a verified ancilla may be created some distance away and need to be moved into contact with the data so the ancilla will accumulate errors from movement operations. In either case, we must be prepared to skip QEC altogether rather than holding up the entire computation.

In this project we consider addressing this problem using ancilla decoding [1], a technique originally devised to address the different problem of slow qubit measurements. As we discuss below, ancilla decoding removes the need for post-selection and thus guarantees that any created ancilla can be used with the data. In this case the ancilla can, in principle, be created in close proximity to the data and QEC performed without any additional movement or waiting and QEC need never be skipped. Depending on the circuit layout and gate errors and timings, this may result in lower overall logical failure rates and hence, in the case of a computation using a concatenated QECC, fewer resources required to achieve a given logical error rate. Since fault-tolerant methods and quantum error correction account for a relatively large amount of the resources used to perform reliable quantum computation, such savings are quite valuable. In this work, we compare the performance (in terms of the logical error rate introduced in a noisy QEC procedure) of ancilla decoding with that of ancilla verification procedures in order to demonstrate scenarios where decoding is advantageous even when measurement is no slower than any other operation.

We initially consider a naive “series” verification scenario, in which ancillas are created and verified until verification is passed (with the data being held in (noisy) memory while additional verifications occur if the first is unsuccessful). This avoids the situation where the QEC fails completely, but, as discussed above, is unrealistic as part of a larger computation, since the duration of the QEC is unpredictable.

We additionally compare decoding to more realistic scenarios for both series and parallel ancilla verification. In order to have a high probability for the ancilla to pass, two attempts are made to create the ancilla in either series or parallel. (This is a physically motivated constraint given the discussion above of the sources of errors.) In the series case, Figure 3.8, the ancilla created either passes on the first attempt or does not. If it does, it is stored until it is needed. If it does not, a second attempt is made. If the second attempt also fails, we skip the QEC process altogether. In the parallel case, Figure 3.7, two ancillas are prepared in parallel. If the one closest to the data passes, it is used. If it does not, the second one is tested. If this ancilla passes, it is swapped (using a set of SWAP operations composed of CNOT gates) with the one that is closest. If it also fails, the QEC process is skipped. In all cases we assume that the ancillary systems are created as close as possible to the data so that the series, first parallel, and the decoding ancilla are all assumed equally close to the data in their respective implementations. While not necessarily applicable to every physical layout, this should be the case for many two-dimensional systems, giving some generality to our results while taking physical constraints into account. We discuss layout further in Section 3.2.4.

3.1.2 The ancilla creation in the Steane code

In this section we compare the logical error rate P_L obtained when performing fault-tolerant QEC operations for [[7, 1, 3]] Steane code [14], using either ancilla verification or decoding. In section 2.9 we studied Steane code and we have shown that Steane code encodes a single logical qubit into the state of seven physical qubits, and can correct any error on a single physical qubit. It is a CSS stabilizer code [13, 20–22], whose list of stabilizer generators is given in Table A.2 in appendix A. In our analysis we consider the Steane ancilla technique for QEC, in which the code stabilizers are measured by copying error information to ancillas in

encoded logical states.

A (non-fault-tolerant) picture of part of the QEC process is shown in Figure 3.1. We see, for example, that for fault-tolerance the $|0\rangle_L$ ancilla, which interacts with the data as the source for a transversal CNOT gate, needs to be prepared so that a single gate failure will not cause the ancilla to have multiple Pauli X errors (likewise with the $|+\rangle_L$ state and Z errors). Such errors would get transferred to the data, as can occur in the circuit shown in Figure 3.1, since the CNOT gates involved in preparing $|0\rangle_L$ can propagate a single gate error to multiple qubits within the ancilla block.

The standard approach for ensuring this using ancilla verification (which occursbetween ancilla preparation and data interaction) is shown in Figure 3.2. Passingthe verification procedure is dependent on the outcome of the transversal Z measurement performed on the verifier. An analogous procedure occurs for Z errors and the $|+\rangle_L$.

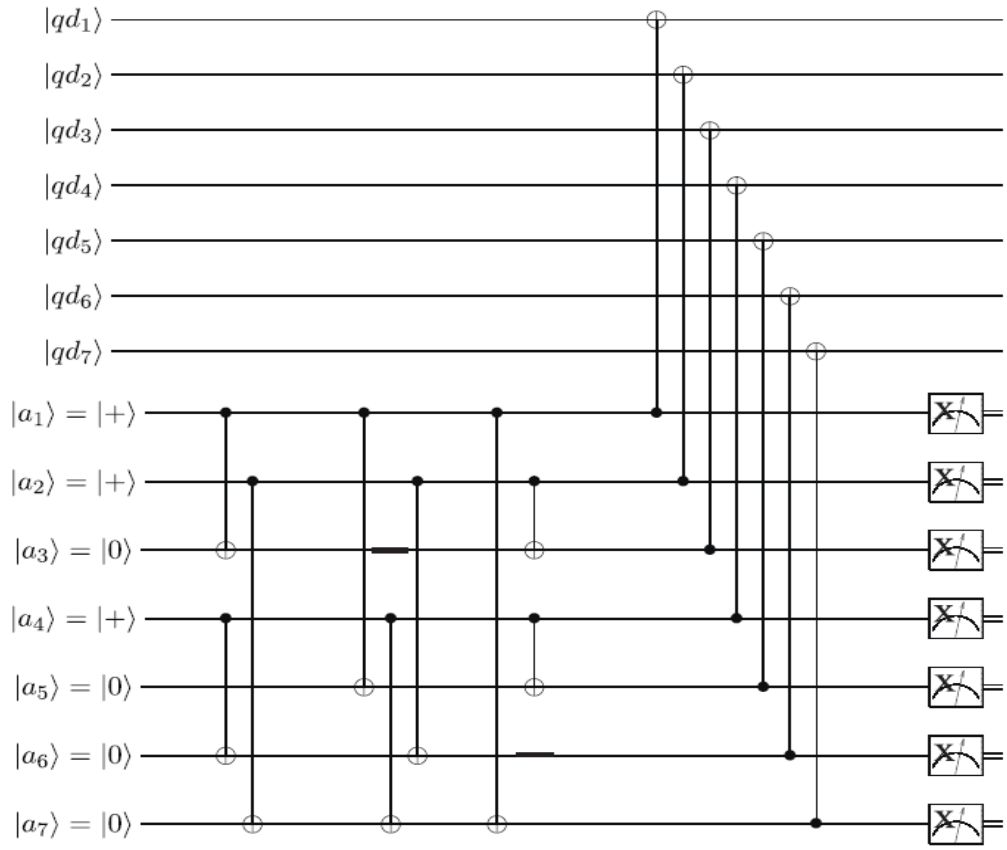


Figure 3.1: A Steane code circuit with non-fault-tolerant Steane syndrome extraction.

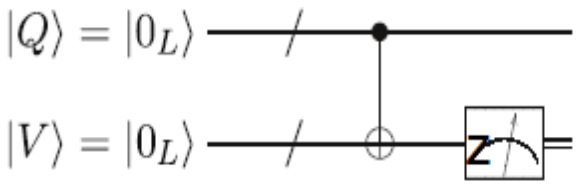


Figure 3.2: An example of a single ancilla preparation and verification circuit taken from the Steane code with Steane syndrome extraction.

The alternative method of ancilla decoding [1] is illustrated in Figure 3.3. The basic principle is that after interacting the ancilla with the data, one applies a decoding operation to the ancilla

to extract both the error syndrome (as usual) and to determine the presence of any errors on the ancilla which may have been transferred to the data (X or Z, depending on the ancilla in question). In addition, the ancilla interacts with a second ancilla block which is in a product of $|0\rangle$ states (i.e. $|0\rangle^{\otimes 7}$), which is also decoded after interacting with the data. This allows one to distinguish between errors of the first ancilla acquired during ancilla encoding (which will have been propagated to the data and need correcting) and those during decoding (which will not be transferred to the data or to the second block). The decoding circuit is simply the time-reversal of the ancilla creation circuit as illustrated for state $|0\rangle$ in Figure 3.1, with the CNOT gates applied in reverse order and creation of $|+\rangle$ / $|0\rangle$ states replaced with measurement in the X / Z bases respectively.

Any propagated errors (single or multi-qubit) can then be corrected on the data. An important factor in the success of this technique is that, for the purposes of fault-tolerance, only “first-order” patterns of errors, which can be caused by a single gate failure (though they may still affect multiple ancilla qubits through error propagation, or through errors on two-qubit gates) need to be corrected. Furthermore, many error patterns are equivalent up to stabilizer operations.

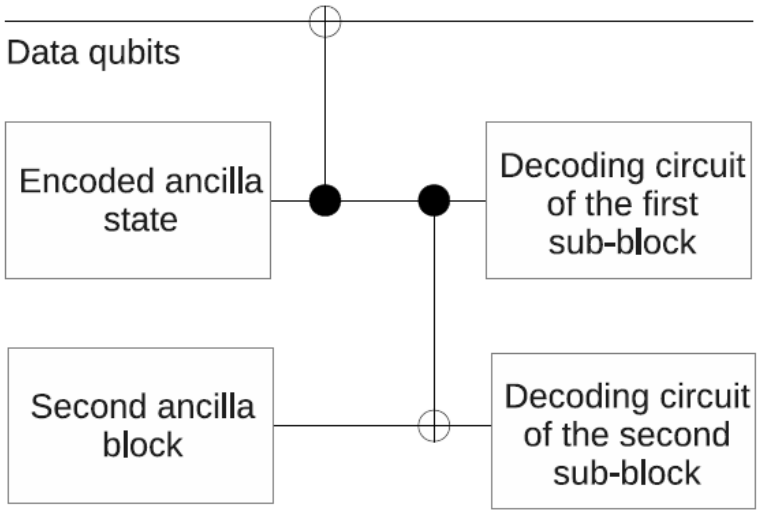


Figure 3.3: The ancilla decoding procedure. After interacting the data with a nonpost- selected ancilla a decoding procedure is applied to determine the error syndrome and any multi-qubit errors which may have been transferred to the data. A second block allows one to distinguish between errors from encoding and decoding

This allows the limited information from the decoder to be sufficient to correct any such errors. Secondly, different classes of errors (X and Z or vice versa) are respectively detected by the standard syndrome measurement and by the additional decoder syndrome. For example, a $|0\rangle$ ancilla is used to determine the syndrome for Z errors on the data, but is at risk of propagating X errors to the data, with the additional decoding giving information about the latter. The use of this method, therefore, means that any ancilla which is created (regardless of first-order errors) can be used in QEC, removing the need for verification. The encoding circuit is illustrated in Figure A.1 in appendix A, each gate in that circuit being faulty and capable of producing an error which will propagate (either as a single or multi-qubit error) to the data via the transversal CNOT gates (as shown in Figure 3.1). For example, if a single fault has been produced by the two output channels of CNOT−7, or −8, or −9 (in Figure A.1), then this error will propagate as a two-qubit error to the data. This is discussed in more detail in appendix A. As mentioned above, ancilla decoding was originally proposed to avoid long waits for ancilla

verification in the case of slow measurements. Since, unless performing non-Clifford-group operations on our data, we can operate in the “Pauli frame” (merely classically recording the necessary corrections and updating the stabilizer accordingly rather than actually applying them), there is no corresponding need for the data to wait for the outcome of the ancilla decoding operations. One could also partially avoid the problem of waiting for verification by simply beginning the verification well in advance, but even for fast measurements, this does not avoid any delays due to having to restart preparation if verification fails or errors due to waiting if the ancilla needs to be stored. In our analysis we consider the case of measurement operations no slower than any other gate, and show that decoding still gives an advantage for this reason, even if verification failure is rare.

As discussed above, we are interested in the limitations imposed by practical QEC rchitectures where a reliable ancilla may not always be available when needed. We simulate cases where ancilla creation occurs serially or in parallel and a failed ancilla verification requires the availability of a replacement, and compare the performance (in terms of the overall P_L) of a QEC procedure under these circumstances to QEC using the decoding method. While we consider the specific case of the Steane code using the Steane ancilla technique, it is conjectured that a similar decoding procedure can be found for other CSS QECCs [1].

3.2 Simulation procedure

To compare methods of ancilla interaction, we performed Monte Carlo simulations of the complete QEC procedure (so interaction of the data with two ancillas, one for the correction of each of X and Z errors) implemented using faulty gate operations.

Our simulation software was QASM-P, software based on QASM by Cross [23]. Initially, all

gates were simulated using the following common stochastic error model for depolarizing noise, a function of a single error probability p:

1. Attempting to perform a data qubit identity I (wait operation), but instead performing a single qubit operation X, Y , or Z, each occurring with probability p/3.
2. Attempting to initialize a qubit to $|0\rangle$ or $|+\rangle$, but instead preparing $|1\rangle$ and $|-\rangle$, respectively, with probability p.
3. Performing a Z-basis or X-basis qubit measurement, but reporting the wrong value with probability p.
4. Attempting to perform a CNOT gate, but instead performing a CNOT followed by one of the two-qubit operations $I \otimes X$, $I \otimes Y$, $I \otimes Z$, $X \otimes I$, $X \otimes X$, $X \otimes Y$, $X \otimes Z$, $Y \otimes I$, $Y \otimes X$, $Y \otimes Y$, $Y \otimes Z$, $Z \otimes I$, $Z \otimes X$, $Z \otimes Y$, or $Z \otimes Z$, each with probability p/15.

(As discussed later, when comparing more realistic series and parallel models we independently adjusted the error rates for the wait and CNOT operations). We considered a range of values of p below the threshold for Steane code of roughly 10^{-4} [16]. All gates are assumed to have the same duration, with wait operations implemented by applying the identity gate the appropriate number of times. To determine PL for the QEC procedure, the data was first prepared, without error, in a logical eigenstate. The QEC procedure was then performed using the error model above. Finally, the data was checked for logical errors. To leading order (where the output data state has no more than two physical qubit errors) we can treat logical X, Y and Z errors on the data as mutually exclusive, thus in this case the overall logical error rate PL can be approximated as $P_X + P_Y + P_Z$ where, e.g., P_X is the probability of the data receiving a logical X error. We determine PL by performing simulations using logical X, Y and Z eigenstates for the data, where, for example $E_X = P_X + P_Y$, where E_X is the simulation-determined logical error rate on eigenstates of the Z basis. From the Monte Carlo

simulation we therefore obtain P_L as a function of the underlying physical error rates.

3.2.1 Simple series simulation

As discussed above, in our simple model for a series procedure of indefinite duration, we assume the ancilla is initially prepared an appropriate length of time prior to the QEC so that the data is not required to wait for ancilla preparation, so long as the first ancilla successfully passes verification. However, if ancilla creation fails, the data is held until a new ancilla can be created and verified. Holding the data continues until verification is passed, at which point the QEC continues. We apply wait operations to the data for a total of 6 time steps (the number required to create and verify a new ancilla) for every failed verification, with no upper bound to the number of failures that can occur. Once an ancilla has successfully passed verification the QEC procedure continues. To determine the extent to which the verification process is affected by this data holding (as opposed to any other element), we additionally simulate an unrealistically optimistic case of verification where no additional holding is required if verification fails.

Additionally, we generated simulation failure rates in which particular classes of errors were considered in isolation, i.e., with other errors turned off. As in, e.g., [24], we define these classes of errors as follows:

- Class 0: errors from preparation and measurement.
- Class 1: errors from single-qubit gates (i.e., wait operations, since correction Pauli gates are error-free).
- Class 2: errors from two-qubit gates (i.e., the CNOT gate).

We additionally performed simulations of more realistic scenarios (preserving data

synchronicity) as described below.

3.2.2 2-ancilla series simulation

In our 2-ancilla series simulation, an ancilla is initially prepared and verified ahead of the data's arrival. If verification fails, a second ancilla is prepared and verified.

To preserve a consistent timing for the QEC operation, interaction of the ancilla with the data with the ancilla always occurs after this second verification would have occurred. Thus, if the first ancilla passes verification it waits for a further 6 time steps before interacting with the data (which is assumed to arrive " just in time" and thus never waits). Hence, unlike the naive simulation, additional waiting occurs in the default scenario (i.e. where the first ancilla passes), and is applied to the ancilla rather than the data. See Figure 3.4.

3.2.3 2-ancilla parallel simulation

In the 2-ancilla parallel simulation, both ancillas are prepared and verified simultaneously. One ancilla (the first) is assumed to be "adjacent" to the data, in the sense that it can interact with the data qubits without any additional movement operations. The second ancilla is considered adjacent to the first ancilla (but not the data). If verification of the first ancilla fails but the second passes, a transversal SWAP operation (consisting of 3 CNOT gates in sequence, the first and third using one qubit as the control and the second using the other qubit) is performed between the two ancillas so that the verified ancilla is now adjacent to the data, after which the QEC proceeds. As before, for timing consistency, the data is assumed to arrive just in time for interaction after this SWAP has occurred. Thus, if the first ancilla passes verification, it undergoes an additional wait operation (the duration of the SWAP gate) for 3 time steps before interacting with the data. See Figure 3.7.

3.2.4 Layout considerations in simulation

We do not model the qubit layout in detail: our simulations (both simple and more realistic) for series verification and decoding do not feature any movement operations. However to roughly simulate the additional errors brought about by ancilla movement in the parallel case, we use the model illustrated in Figure 3.5. Qubits for each logical block are positioned in “interaction regions” (the horizontal rectangles). Within the same region any two qubits may interact without any additional movement. However, for two logical blocks to transversally interact, they must be in adjacent regions (we model the interaction regions as having limited capacity otherwise we could simply place every logical block in the same region in this case, equal to one logical block (7 qubits)). Thus the series and decoding approaches do not require any additional movement operations; as illustrated, we may prepare the “primary” ancilla (which interacts directly with the data) in the region adjacent to the data, and the verifier ancilla (or additional ancilla for decoding), which only needs to interact with the primary ancilla, in the region on the far side of the primary ancilla. We do this on one side of the data for the QEC used to correct X errors, and on the other for the QEC for Z errors, since we need to begin preparation of the ancilla for one QEC while the other is still taking place, so the data requires no additional waiting between these operations.

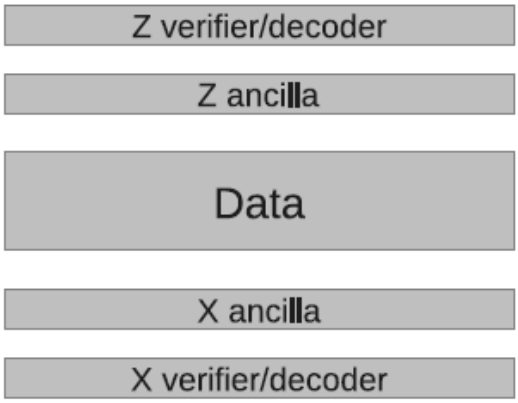


Figure 3.4: Underlying layout for simulations. Ancillas are created in separate interaction regions adjacent both the the data and additional verification or decoding ancillas.

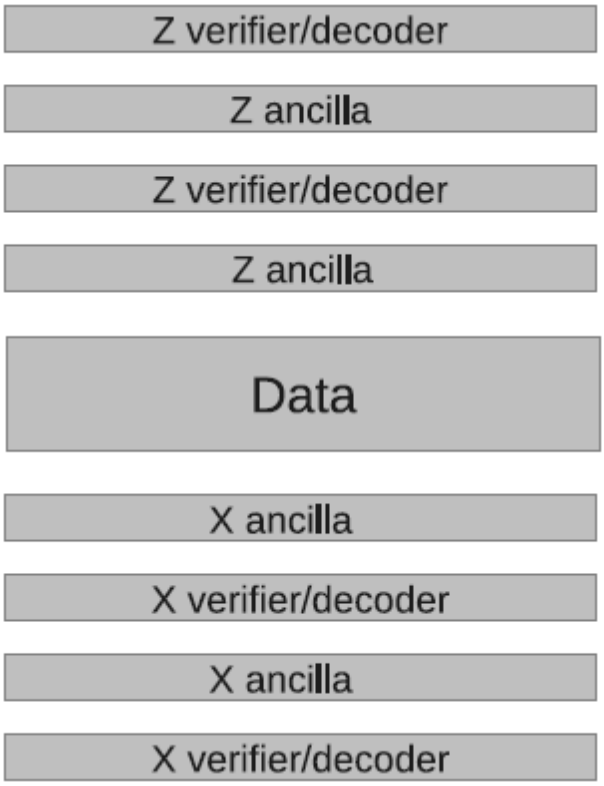


Figure 3.5: Layout for parallel ancilla verification. If ancilla 1 fails verification, ancilla 2 must perform two SWAP operations to reach the data.

Layout in parallel verification In the parallel verification case, if we are permitted one logical block per region, we arrange the ancillas as illustrated in Figure 3.5. Thus secondary ancillas must be moved to interact with the data, giving additional errors. We implement “movement” of this kind by applying SWAP gates (consisting of 3 CNOT gates) between adjacent blocks. Thus, to “move” a secondary ancilla adjacent to the data we must apply two SWAP gates to move it past both the original ancilla and the original verifier. However this means in our model that parallel verification can easily be seen to be worse than the series case, since the two SWAP gates consist of 6 total operations, or the same amount of time as required to create and verify a new ancilla in the series case. Thus the time required post-failure to have a new ancilla available is no less, and additional errors are incurred from the SWAP gates, so the series method is clearly better in terms of errors incurred. Since we are interested in demonstrating where decoding improves on the (best available) verification methods even for fast measurements, we note this result and do not perform explicit simulations for this parallel case.

This result is, of course, dependent on our layout choices and our chosen method for moving the data and second ancilla adjacent. However, it illustrates the cost of parallel methods for ancilla creation - in this case, although not necessarily always, so high as to make series verification uniformly better - of having to move successfully verified ancillas into place. For the series and decoding scenarios which we do simulate, the difference between techniques consists primarily of additional wait and CNOT operations, the former in additional ancilla waiting for synchronicity, the latter in decoding operations, and relative performance should therefore depend primarily on the error rates of these operations. We therefore performed simulations for both scenarios and for ancilla decoding over CNOT and wait error rates ranging from 10^{-5} to 3×10^{-4}, with all other gates fixed at an error rate of 10^{-5}.

There is also, in the series case, the possibility that both ancillas will fail verification and no QEC occurs (which never occurs with decoding). This is obviously a bad outcome (since, in a larger computation, any prior error on the data would not be corrected). However, it does not directly correspond to a logical error induced by the QEC. Since we are interested in demonstrating that decoding can be advantageous even when measurement is not slow, we simply rerun any such scenarios, which do not contribute to final pass/fail statistics. Qualitatively, then, our use of P^L slightly overestimates the performance of the verification techniques in this respect.

3.3 Results and analysis

3.3.1 Simple series simulation

A direct comparison of the PL for all three QEC techniques (decoding, verification, and a naive verification without additional waiting) is shown for the simple series simulation in Figure 3.6. The results show an advantage for the decoding technique over verification over the whole range of errors considered, larger at larger error rates with a roughly 2-fold reduction as the maximum. Comparison with the PL values for verification without additional waits, which are lower still, indicates as expected that the increased errors in the verification technique are due to ancilla failures, and that in the absence of such failures the additional operations required for ancilla decoding lead to a slightly higher error rate. The rate of ancilla failure as a function of p is plotted in Figure 3.7. Since ancilla verification detects single errors on the ancilla, this is roughly equal to (number of error locations in ancilla creation circuit)×p, and the rate scales linearly with p as expected. Thus at low physical error rates ancilla failure occurs in a proportional ly low fraction of QEC events, but still contributes significantly to the logical error rate.

Comparing the error rates by class in Figure 3.8 we see that the largest source of error in verification is due to wait operations, while the other two techniques are dominated by CNOT errors. This emphasizes the need to take into account non-deterministic operations (such as verification) when assessing the impact of gate errors on QEC procedures.

The technique, over a range of CNOT and wait error rates. In qualitative terms, these generally are as expected, but importantly show when decoding is the better technique. We see again that decoding outperforms verification (i.e. has a lower logical failure rate) for a large range of error rates. In contrast with our simple series simulation where all gates had the same error rate (and decoding was consistently superior), we see, however, that verification outperforms decoding in the range of low CNOT error and large wait error. This is as expected, given that, relative to each other, verification requires additional wait operations (to recreate ancillas) and decoding additional CNOT operations (to perform decoding). As discussed in section 3.2.4, the parallel approach will be uniformly worse than the series approach in this model, so where decoding outperforms series verification it will also outperform parallel verification.

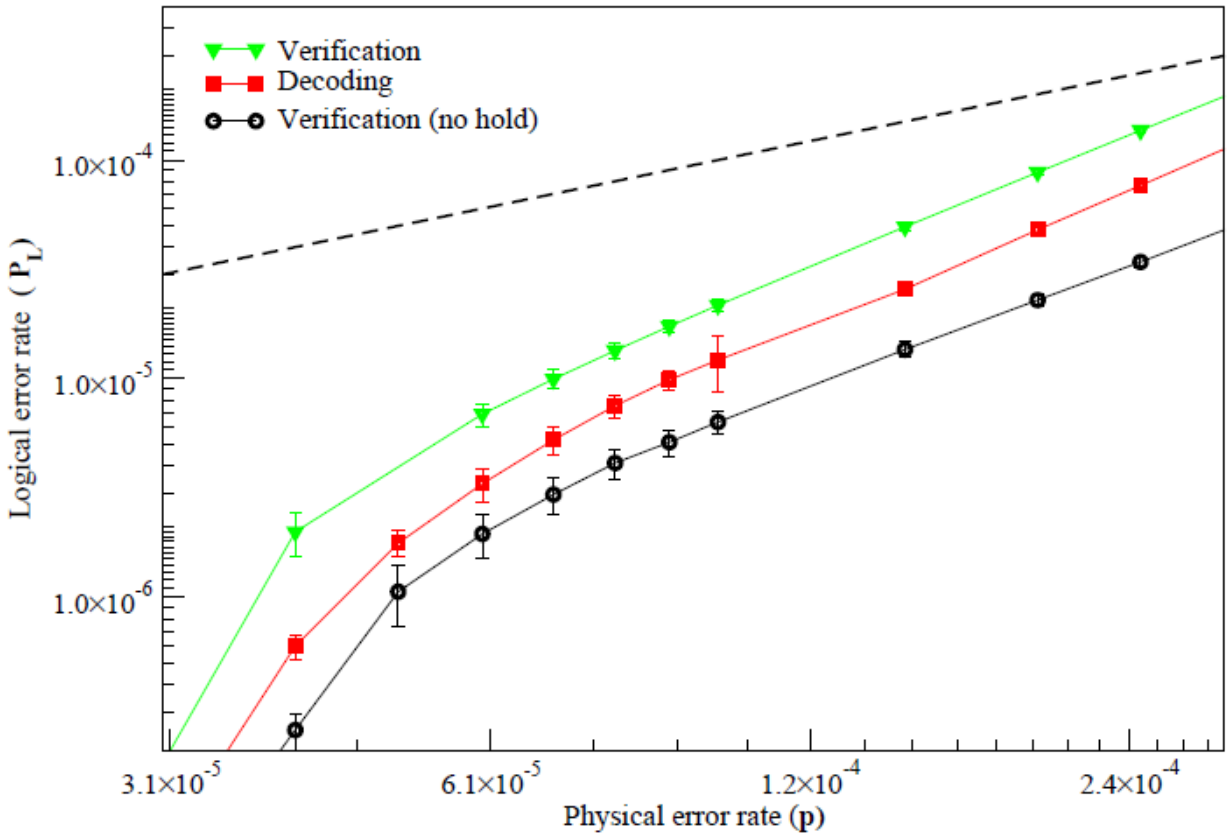


Figure 3.6: Logical error rate as a function of physical error rate for three QEC techniques. Dashed line represents p = PL.

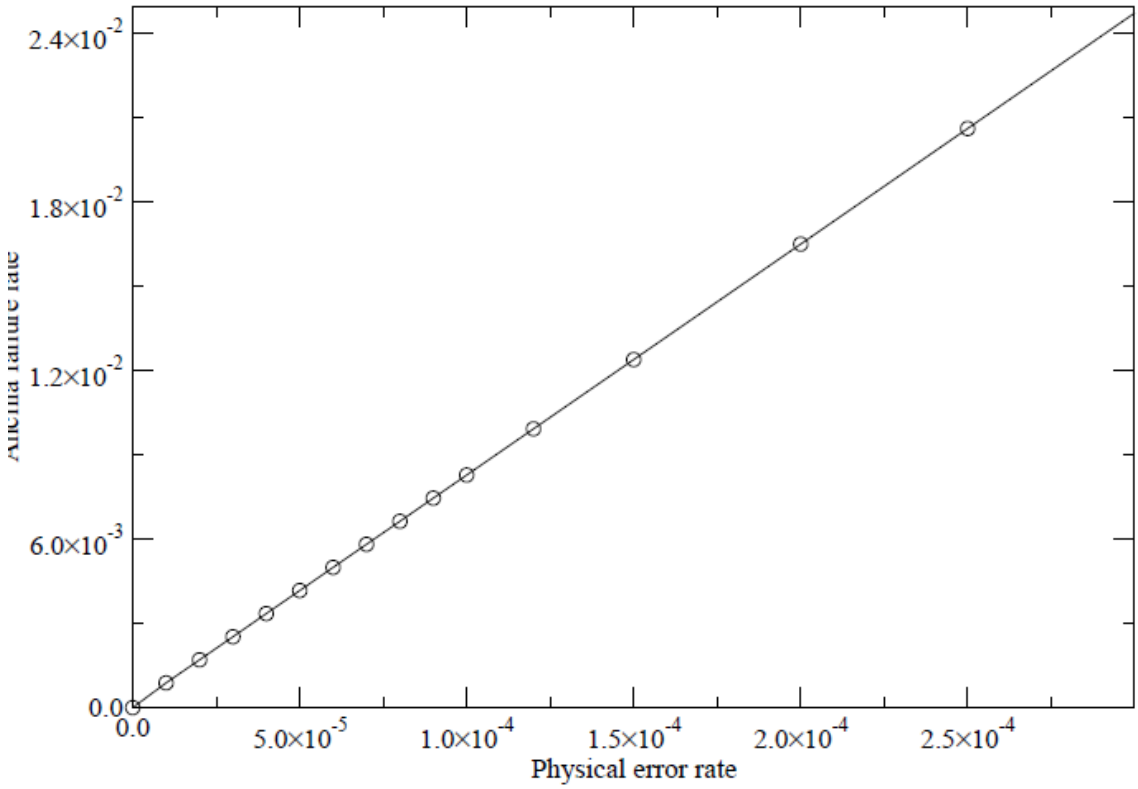


Figure 3.7: The ancilla failure rate vs. physical error rate for the [[7,1,3]] Steane code with Steane ancilla

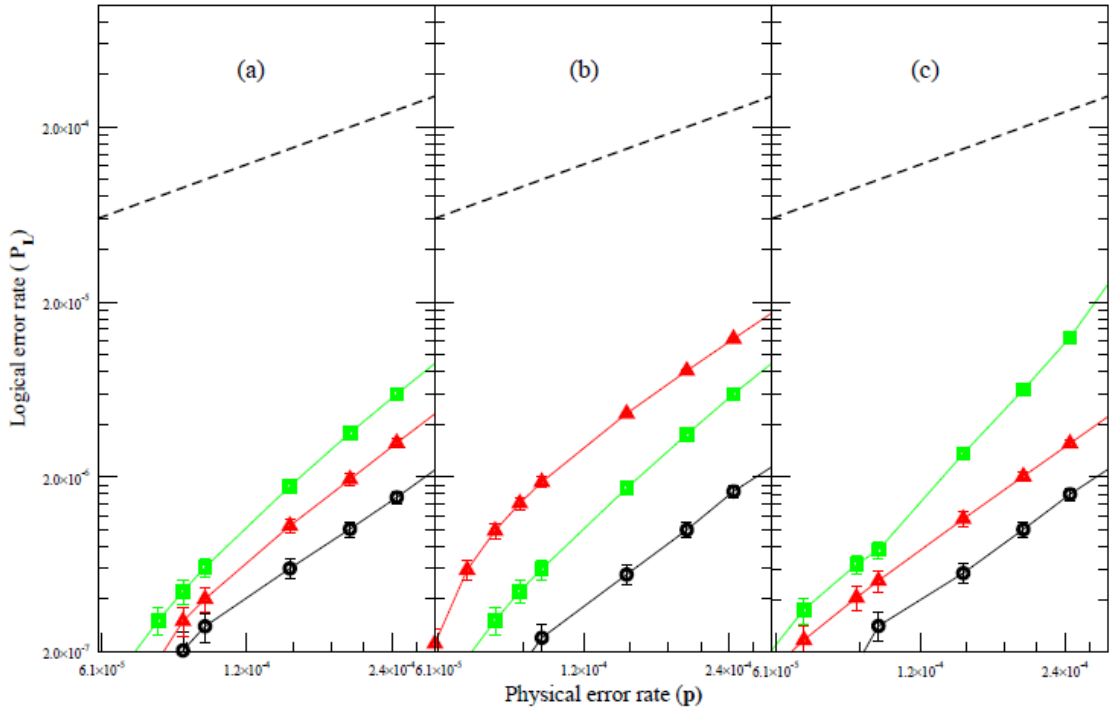


Figure 3.8: Logical error rate PL vs. physical error rate p for class-0 errors (black circle points), class-1 errors (red triangle points), and class-2 (green square points) for (a) verification (without additional waits), (b) verification and (c) decoding. Dashed line represents p = P_L.

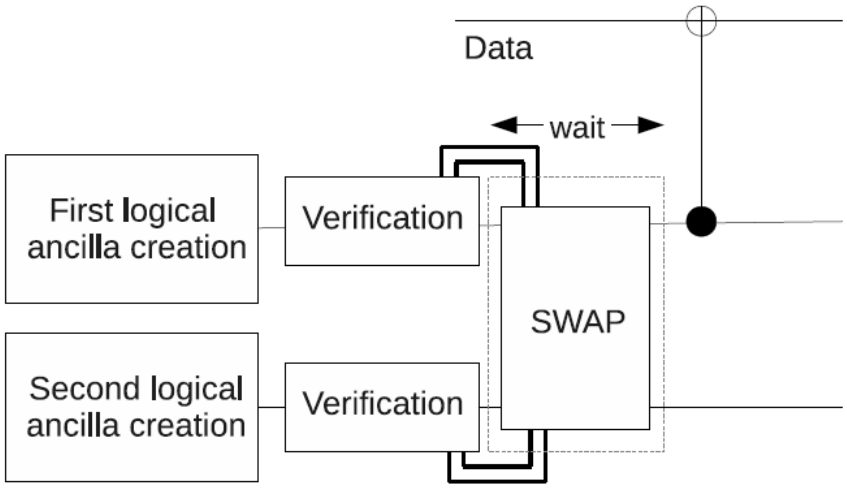


Figure 3.9: Parallel ancilla verification. If the first logical ancilla state has passed the verification test, it waits for 3 time steps (the duration of a SWAP gate) before interacting with the data (example shown is for correction of Z errors). Otherwise (if only the second ancilla passes verification), a SWAP operation is applied between two states, and the verified ancilla used for QEC. Operations in the dashed box are conditioned on the first verification, either they or the WAIT operation takes place. If no verifications pass the QEC is abandoned.

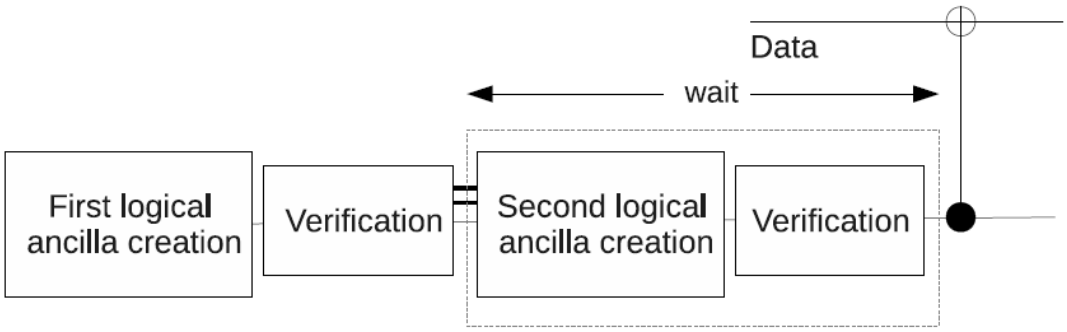


Figure 3.10: Series ancilla verification. If the first logical ancilla state has passed the verification test, then, it waits for 6 time steps (duration of ancilla creation and verification) before interacting with the data. Otherwise (if verification fails), we prepare and verify the second logical ancilla state and use that for QEC if verification passes. Operations in the dashed box are conditioned on the first verification, either they or the WAIT operation takes place. If no verifications pass the QEC is abandoned.

CHAPTER 4

OPTIMAL FREQUENCY FOR THE APPLICATION OF QUANTUM ERROR CORRECTION USING THE [[7,1,3]] STAENE CODE

4.1 Introduction

The standard circuit model of quantum computing breaks down operations into individual quantum gates, the physical implementation of which can generally only be done imperfectly, leading to some probability of errors on the quantum data. To protect against these errors, logical qubits are commonly encoded into multiple physical qubits using a quantum error correcting code (QECC) [2, 13, 14, 25], using which additional quantum error correction (QEC) operations are regularly performed to diagnose and repair errors that may have arisen.

A common model for such an error-corrected computation is to apply a QEC immediately after each gate, in order to correct the physical errors introduced by these gates before they accrue into more serious logical errors. However, QEC operations will, in general, also introduce errors with some non-zero probability since they also require imperfect physical gates (often the same ones as are used to perform logical operations) to implement. For QEC operations to successfully suppress errors enough to allow large-scale computation, they (as well as logical gates) must be implemented in a fault-tolerant way [4], i.e., in a way such that a single faulty quantum gate cannot lead to multiple errors on the data.

Fault-tolerant constructions for QEC operations are often dependent on post-selection; in particular, they commonly use ancillary states which are used to diagnose errors, effectively to carry away entropy from the data. Since they interact with the data, these states must be prepared so that preparation errors do not spread multiple errors to the data, and this preparation must be performed fault-tolerantly. Often, however, non-fault-tolerant circuits must

be used for the initial ancilla preparation, and fault-tolerance is instead enforced by post selection (ancilla verification) in which only those ancillas which satisfy some measurement outcome after being created are subsequently used to interact with the data [13, 14].

This model of fault-tolerant QEC, in which ancillas are created until one passes post-selection, can be difficult to implement: the data can accumulate additional errors either waiting for a "good" ancilla to be created (if created sequentially) or to be moved into place (if created in parallel). And the nondeterministic delays involved make synchronisation of the data with other data blocks difficult. An alternative, which preserves synchronisation, is to simply to allow for a fixed number of postselection attempts, and to carry on with the computation (skipping the QEC) if these are unsuccessful. The obvious disadvantage is that skipped QEC operations allow errors to accumulate from sequential logical gates, but if the skipping probability and gate error probabilities are sufficiently low this may still be the optimal solution.

A closely related situation is where QEC operations are intentionally not applied after every gate, but only after a certain number of gates. This can reduce the overall logical error if the errors introduced by the QEC operation are larger than those introduced by the gates. Recently, Weinstein has provided a specific relation between the fidelity and physical error rates for performing QECs after different numbers of gates in the [[7, 1, 3]] Steane code [26]. Here, it is shown that the overall error rate is not significantly increased by performing QEC after two gates rather than after every gate, and that the inherently noisy correction process will sometimes introduce more error into the system when applied too frequently [26,27].

In this paper, we consider the optimal frequency to apply postselection-dependent QEC operations, in the case where postselection failure results in a skipped QEC.

We consider a model where a logical data qubit undergoes operations from N logical

gates, with m gates in between each QEC, using the well-known [[7,1,3]] Steane code [14] and the Steane ancilla technique for the latter. We determine the logical error rate PL for the data after undergoing these (N/m) “blocks” of gates and QEC operations, as a function of m and the underlying physical gate error ϙ, and then minimise as a function of m.

In section 4.2 we explain the model and the analytical derivation of the formula given the model. In section 4.3 we compare the predicted error rate of the model to that obtained via Monte Carlo simulation.

In order to reduce the occurrence of errors, a logical qubit is encoded into a block of seven physical qubits, likewise, an operation applied to a logical qubit must be implemented by gates (i.e. physical operations) which are naturally fault-tolerant. An example of this kind of gate is a transversal gate, in which every physical qubit is acted on by a separate gate such that there is no opportunity for errors to spread within that logical qubit block. The [[7,1,3]] Steane code used here is able to restore a logical qubit to its proper state when a fault exists in just one of these physical gates causing an error to one of the seven physical qubits, these gate faults occur at physical error rate . Therefore, the information will suffer a logical error only when two or more errors occur within a set of seven physical qubits making the logical error rate $P_L \sim O(\epsilon^2)$ [2].

4.2 Derivation of the mathematical formula for P_L

4.2.1 Logical gate and QEC model

In general, the dependence of the logical error rate on the physical error rate of the underlying gates depends on the circuits used to implement gates and QEC. These circuits can be complex, and highly dependent on the chosen code. While our analysis is based on the Steane code, we wished to use a model that could be readily adapted to other codes. We thus

produced a semi-abstracted model based on the Steane code with Steane ancillas. Errors may be introduced into the logical qubit in two ways, from the logical gates and the " noisy" QEC itself (in this analysis we do not treat errors from movement or hold operations as a separate category, they may incorporated into the above categories if desired). We model a noisy physical gate as performing the desired operation followed by, with probability ϵ_g, an error operation. We treat logical gates as transverse, that is, simply consisting of a single physical gate applied to each qubit.

Our model of the QEC operation is more approximate. We divide errors induced by QEC into three separate parts with the following probabilities:

- "Syndrome errors", with probability ϵ_s per qubit, are errors applied to the data due to an intentionally applied, but erroneous correction operation (for example, due to errors in the syndrome measurement process). Additionally, we assume that when such an error is present, the QEC does not successfully correct any existing errors on the data. Multiple syndrome errors within a single QEC will (for distance-3) only result in one error on the data.
- "Correction errors", with probability ϵ_c per qubit, are due to physical errors in those gates directly applied to the data as part of the QEC process. In the Steane code and many others, this is limited to those two-qubit gates used to interact the data with the ancillas (since the corrections themselves can be done using the "Pauli frame", without the use of physical gates).

We additionally consider the probability of an "ancilla error", with probability ϵ_a (not per qubit). This is the case where a QEC operation (that is, any elements of that operation that could either correct errors or produce errors on the data) is not performed (for example, due to failure in the postselection process for the necessary ancilla). Such events do not produce data errors

themselves, but result in existing errors not being corrected when they should be.

Note that all of the above are functions of the physical gate error ǫg, but the exact relationship depends on the circuits and QECC used, thus we treat them as separate variables in our model.

4.2.2 Logical error rate

The sequence of (N/m) blocks may, in a distance-3 code, produce a logical error if one or more of these blocks create two or more physical errors on the logical qubit. Therefore, to estimate the logical error rate PL, wemust enumerate the ways in which these blocks might create two or more physical errors on the logical qubit. This logical error might be created either within a single block, or across multiple blocks (if a QEC is either skipped due an ancilla error or fails due to a syndrome error. Thus a logical error will only occur with probability second-order or higher in the syndrome, correction or gate error probabilities.

We are particularly interested, however, in the regime where the ancilla errors are significantly larger than the other errors. This can easily be the case if ancilla creation circuits are complex. While the postselection procedure means such large errors do not translate directly to logical errors, they can result in multiple QEC operations being skipped, with a consequent increase in the logical error probability if other errors are present. Considering now the structure of a block, this consists of m gates, followed by a QEC operation consisting of a successful correction of any errors on the logical qubit, unless a syndrome error occurs at this point. Finally any correction errors are applied. Thus, in the absence of QEC verification failures, the leading-order contributions to the logical error rate (where two errors occur resulting in a logical error) are very limited: two errors can occur within a block (from two gate errors or a combination of gate and syndrome errors) or across two adjacent blocks (one gate error from each block, a gate error from one block and a syndrome error from the other, one syndrome

error from each block or a correction error on one block followed by an error of any kind on the second block).

A verification failure permits additional error combinations: errors on two separate blocks between the locations which a successful correction would ordinarily occur. Note that additional verification failures permit, to second order, the same types of logical errors. That is, the only second-order errors which f skipped QECs additionally permit (beyond those which could occur regardless) are when the skipped QECs all occur sequentially, and the two errors in question are on the block containing the first skipped QEC and the block following the final skipped QEC.

For a sequence of B $\equiv$ N/m blocks, there are B − f ways to have f sequential skipped QECs. Thus the additional logical error due to f skipped QECs is weighted by a factor

$$\gamma = \sum_{i=1}^{B-1} \epsilon_a^f \ (B-f) = \frac{B\epsilon_a(1-\epsilon_a)-\epsilon_a+\epsilon_a^{B+1}}{(1-\epsilon_a)^2} \tag{4.1}$$

Due to the definition of correction errors, in some cases the combined errors allowable due to a single skipped QEC may span 3 rather than 2 blocks, in which case the multiplying factor is

$$\gamma = \sum_{i=1}^{B-1} \epsilon_a^f \ (B-f-1)$$

$$= \frac{\epsilon_a}{(1-\epsilon_a)^2}(\mathrm{B}(1-\epsilon_a)-2+\epsilon_a+\epsilon_a^{B-1}) \tag{4.2}$$

In Table 4.1, we show the possible ways for a logical error to occur to second order in ϵ_g, ϵ_c and ϵ_s. The final column represents the overall contribution to the logical error rate from all error combinations of that type. Note that rows represent a general class of errors (e.g. the first row represents the contribution due to two gate errors within a single block) and the rows involving factors of γ and γ_3 represent classes of errors where intermediate QEC steps are skipped. These, although only showing a single skipped QEC between the errors in the table, also represent analogous situations with multiple intermediate skipped QECs, leading to an overall factor of of γ and γ_3. Summing the terms, the general second-order formula for P_L is :

$$P_L = 42\left[B\left(\frac{(m\epsilon_g)^2}{2} + m\epsilon_g\epsilon_s + + \frac{{\epsilon_c}^2}{2} + \epsilon_s\epsilon_c\right) + (B-1)\left(m\epsilon_g(\epsilon_c+\epsilon_s) + {\epsilon_s}^2 + \epsilon_s\epsilon_c\right) + \gamma\left(\left(m\epsilon_g\right)^2 + m\epsilon_g\epsilon_s\right) + \gamma_3(m\epsilon_g(\epsilon_c+\epsilon_s) + \epsilon_s\epsilon_c + {\epsilon_s}^2)\right] \quad (4.3)$$

4.3 Monte Carlo Simulation of P_L

As discussed above, our analytical formula for the logical error simplifies the description of QEC errors (in general a function of complex ancilla circuits) to the variables ϵ_c and ϵ_s, which we assume to be the same for every qubit. In order to check the accuracy of this approximation, we performed Monte Carlo simulations of the complete QEC for the [[7,1,3]] Steane code with the Steane ancilla technique, using QASM-P, simulation software based on QASM [23], in order to compare the logical error rates obtained with those predicted. Initially, all gates were simulated using the stochastic error model for depolarizing noise [28]. In this case, we considered bit-flip (X) errors only on the data qubits (phase-flip (Z) errors may be dealt with independently in the [[7,1,3]] code, and in our chosen noise model, will occur at equal rates). We used N = 1000 with varying block sizes m $\in$ {1, 2, 4, 5, 8, 10, 20, 25, 100}, thus B varied between 1000 and 10.

Ancilla verification is performed as usual for X correction using the Steane technique in the

[[7, 1, 3]] code: an ancilla interacts with the data as the control for a transversal CNOT gate, and needs to be prepared and verified so that a single gate failure will not cause the ancilla to have multiple errors which would be transferred to the data. Usually, then, ancilla verification failure rates are, like the other errors, a function of the gate error rates. However, in order to vary ϵ_a independently of other errors to verify the formula, our simulation artificially determines beforehand whether a verification failure error will occur. If so, the QEC is skipped. If not, the preparation and verification is repeated until passed. Thus verified ancillas have the correct error statistics for a given gate error, but failure occurs with the chosen probability ϵ_a. To determine the logical error probability P_L, the data is prepared, without error, in a logical $|0\rangle$ state. Then, as described above, a series of blocks of m transversal logical gates (since we only wish to simulate errors, these are simply wait operations which in the absence of errors do not change the qubits' state), followed by one QEC operation per block, are applied, for a total of N logical gates and B = N/m blocks and (attempted) QECs. Finally, the data is checked for logical X errors.

Each simulation (for a given choice of variable values ǫg and ǫa has 106 runs). From the Monte Carlo simulation we therefore obtain P_L as a function of the underlying physical error rates.

4.3.1 Numerical estimation of ϵ_s and ϵ_c

of ϵ_s and $\epsilon_c$can be determined either by analysis of the QEC circuits or direct measurement via simulation.

ǫs and ǫc were determined directly from simulation. By our definition, a syndrome error in a QEC will, for an input logical qubit containing one error, add a second error (unless the syndrome error coincidentally affects the same qubit as had the original error), leading to an overall logical error, and is the only first-order QEC error which does this. Thus to estimate

error rate of ϵ_s, the data was first prepared in a logical eigenstate, with an error on one of the seven qubits. The QEC procedure for the [[7,1,3]] Steane code with the Steane ancilla technique was then performed using the stochastic error model for depolarizing noise [28]. Finally the logical qubit was checked for logical errors to determine of ϵ_s.

On the other hand, if a correction error occurs in a QEC where the input data has one error, then the original error will be successfully corrected in addition to the correction error being applied (or, if affected qubits are coincidentally the same, the original error will simply be left uncorrected). Thus a single input error leads to a single output error. Hence to estimate the QEC physical error rate of ϵ_c, we prepare the input logical qubit with an error on one of the seven corresponding physical qubits, then we perform the same QEC simulation procedure as in the case of of ϵ_s, but determine the rate based on events when the output has a single error (rather than two as in the case of of ϵ_s). In both cases we varied the input error over all 7 qubits and took the mean resultant P_L.

We performed the simulation with a variety of numerical values for of ϵ_g (10~ ~ 10^{-4}). For each different value of ϵ_g, we determine the numerical values of ϙs and ϙc The relationship is fitted to a linear equation and we used the slopes of the best fit straight line for ϵ_s vs. ϵ_g and ϵ_c vs. ϵ_g, which were as follows:

$$\epsilon_s = 3.02\,\epsilon_g \qquad (4.8)$$

$$\epsilon_c = 1.034\,\epsilon_g \qquad (4.9)$$

4.4 Results and discussions

Figures 4.2, 4.3, and 4.4 show the relationship between P^L and m for the case ϵ_a = 0.0, 0.3, 0.5, respectively, for gate error values ϙg = 5.0 $\times$ 10^{-5}, 8.0 $\times$ 10^{-5}, 1.0 $\times$ 10^{-4}, and 3.0 $\times$ 10^{-4}. The black circular points are the values for PL determined from the numerical

simulation (section 4.2), the red square points are for a simplified version of PL in the limit B → ∞, and the blue triangular points are for PL which are determined by using the exact formula. Our primary observations are that there is generally good agreement between the data generated by the formula and the simulation (and reasonably good agreement even given the assumption of large B, especially with respect to the location of m_{min}), and that m_{min} is insensitive to variations in ϱa over the range of ϵ_a considered, with m_{min} = 5 in all cases. As expected, within the region m < m_{min}, the error rate is reduced both by increasing m and by increasing ϵ_a, since both result in fewer QEC operations being performed (the only difference being whether the skipped operations are regularly spaced or not), and (for given m_{min} QEC operations in this region produce more errors, on average, than they correct. Similarly the behavior is reversed for m > m_{min}.

The largest disparity is for ϵ_a = 0 and small m. Out of all the regions we have considered, this is where the largest number of QEC operations occur (since they are attempted frequently and none are skipped), indicating that the approximations in modeling QEC errors are leading to a significant underestimate of the overall logical error. At higher values of m (where gate errors are a more dominant source of error) agreement is much better. We also find that the assumption of large B (which we use in the analytical derivation of m_{min} gives reasonably good agreement over most of the ranges considered, with the exception of the ϵ_a = 0, small m discussed above and, as expected, where m is large and the approximation is no longer valid

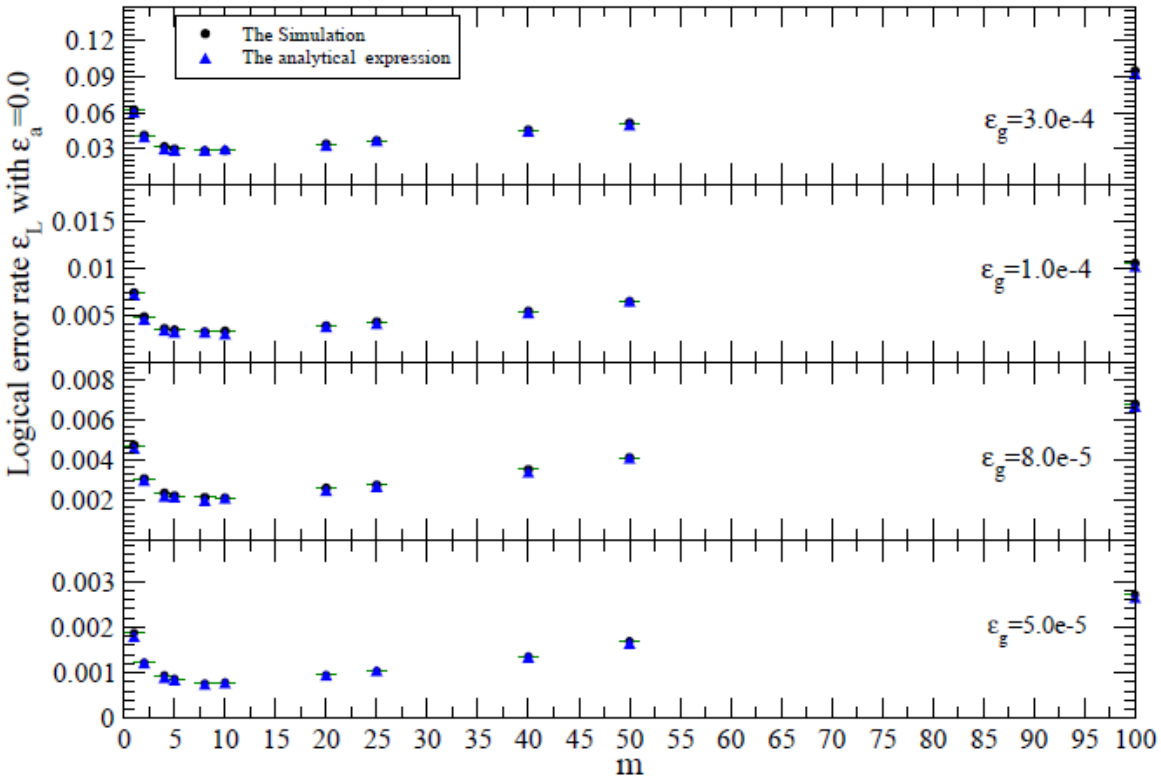


Figure 4.2: P_L vs m for ϵ_g= 5.0 × 10^{-5} ,8.0 × 10^{-5}, 1.0 × 10^{-4} , 3.0 × 10^{-4}, where ϵ_a for this case is zero.The triangular blue points are the numerical values of P_L as shown in the equation (4.3) . The circular black points are the numerical simulation values of P_L. the red square points are the values of the PL using a simplified formula for the limit B → ∞.

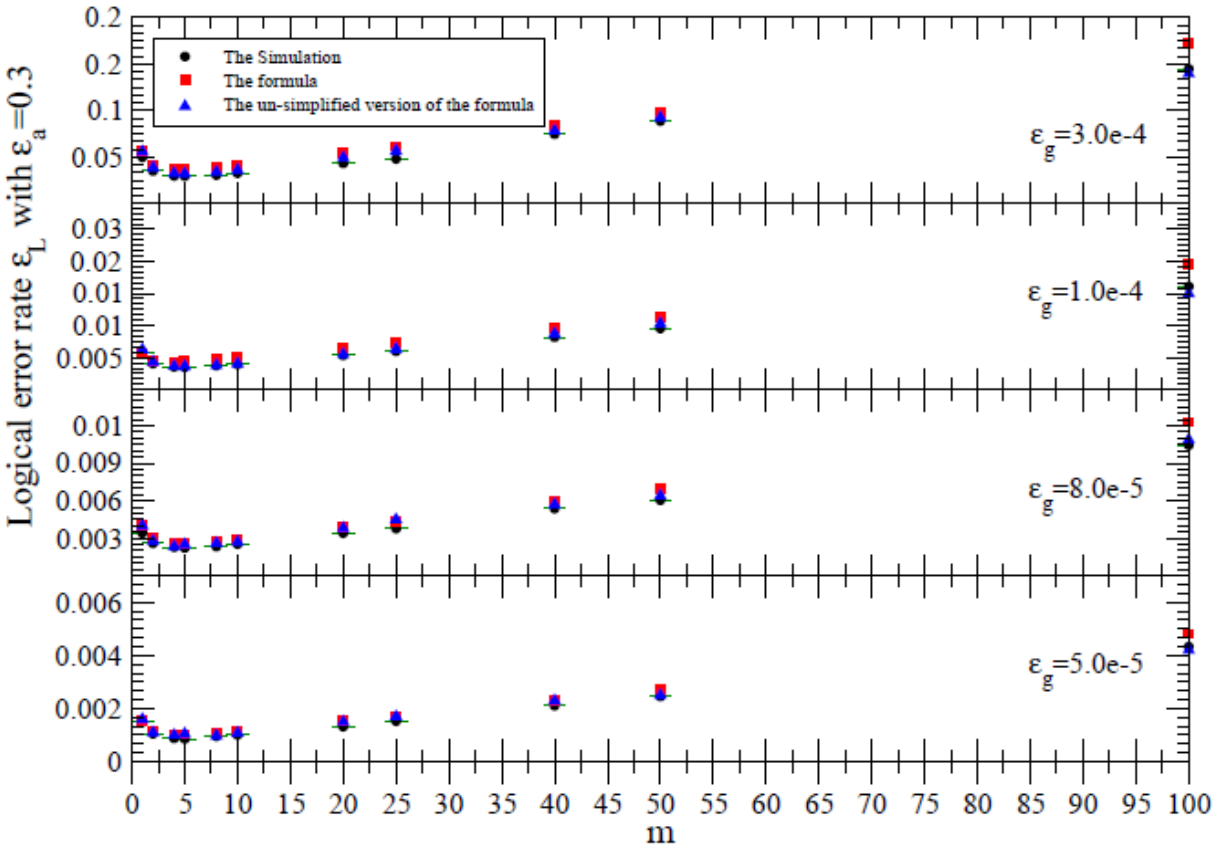


Figure 4.3: P_L vs m for ϵ_g= 5.0 × 10^{-5} ,8.0 × 10^{-5}, 1.0 × 10^{-4} , 3.0 × 10^{-4}, where ϵ_a=0.3.

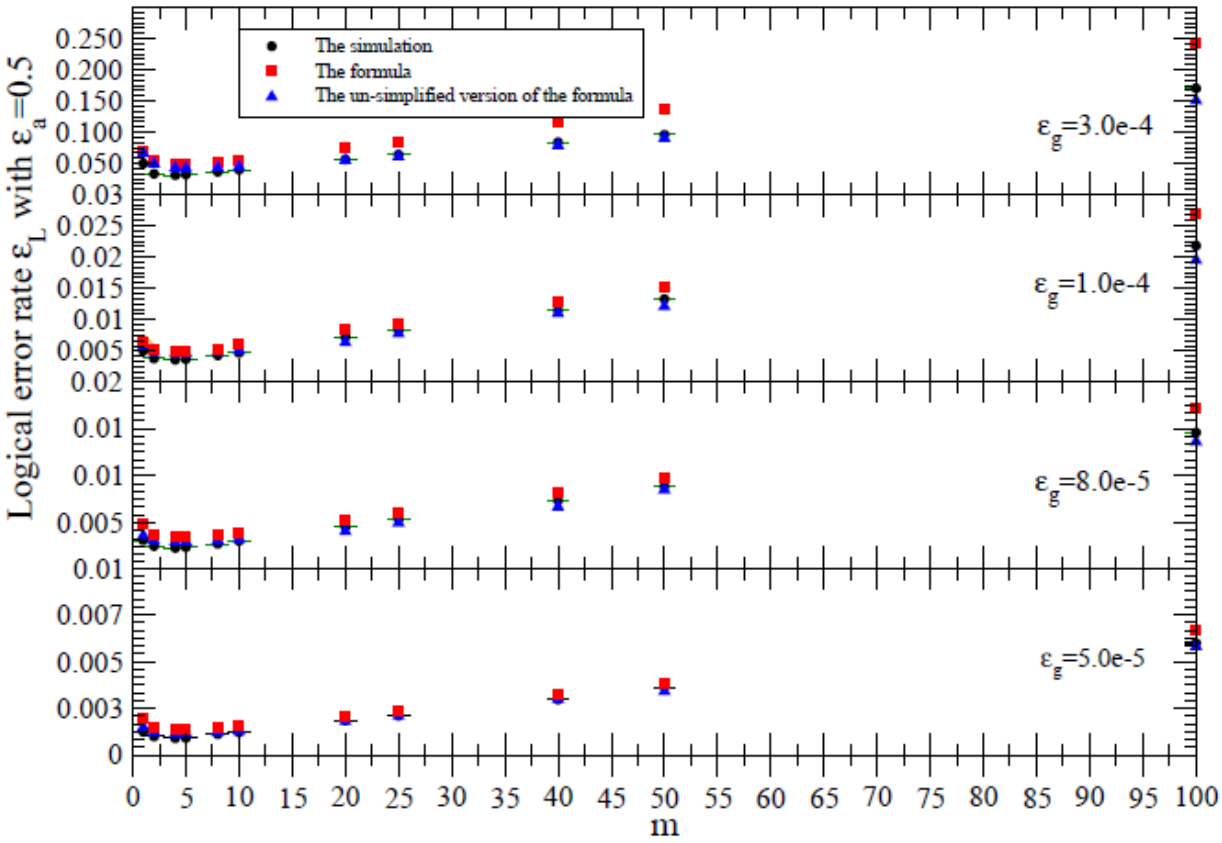


Figure 4.4: P_L vs m for ϵ_g= 5.0 × 10^{-5} ,8.0 × 10^{-5}, 1.0 × 10^{-4} , 3.0 × 10^{-4}, where ϵ_a=0.5.

Table 4.1: This shows all possible second-order errors, which can occur across up to 2 blocks in the absence of skipped QECs, or up to f + 2 blocks for f QECs. Entries indicate the number of errors occurring in a particular location. The contribution indicates the total contribution to the logical error from all errors in the representative class, which includes all possible (non-zero) numbers of skipped QECs for contributions involving a factor γ or γ_3. mG denotes a set of m gates, $\epsilon_s$and ϵ_c the syndrome error and correction error elements of a QEC operation. X denotes an operation omitted due to a skipped QEC.

Error source									Error contribution
mG_i	$Q_i(s)$	$Q_i(c)$	mG_{i+1}	$Q_{i+1}(s)$	$Q_{i+1}(c)$	mG_{i+2}	$Q_{i+2}(s)$	$Q_{i+2}(c)$	(overall factor 42)
2									$B(m\epsilon_g)^2/2$
1	1								$Bm\epsilon_g\epsilon_s$
	1	1							$B\epsilon_s\epsilon_c$
		2							$B(\epsilon_c)^2/2$
	1		1						$(B-1)\epsilon_s m\epsilon_g$
	1			1					$(B-1){\epsilon_s}^2$
		1	1						$(B-1)\epsilon_c m\epsilon_g$
		1		1					$(B-1)\epsilon_c\epsilon_s$
1	X	X	1						$\gamma(m\epsilon_g)^2$
1	X	X		1					$\gamma m\epsilon_g\epsilon_s$
	1			X	X	1			$\gamma_3 m\epsilon_s\epsilon_g$
	1			X	X		1		$\gamma_3(\epsilon_s)^2$
		1		X	X	1			$\gamma_3\epsilon_c m\epsilon_g$
	1			X	X		1		$\gamma_3\epsilon_c\epsilon_s$

CHAPTER 5

CONCLUSION

In chapter 3, we have used the Monte-Carlo Simulation to compare the logical error rate, P_L , for implementations of the Steane code with Steane ancillas using dfferent QEC techniques: ancilla verification with serial and parallel production of multiple ancillas, and the ancilla decoding procedure. We find that, even when measurement times are no longer than those for other operations and verification failures are rare, the decoding procedure, though otherwise more complex, is often advantageous in avoiding additional data waiting and/or movement due to verification failures and leads to lower failure rates.

As the data shows, the relative performance of the various techniques is highly dependent on the underlying error rates of the individual operations. A model where gates have equal errors, ancilla creation continues serially until success and all logicl blocks are assumed adjacent to those with which they interact consistently produces higher logical error rates for verification than for decoding, but, for example, small wait errors and large CNOT errors can result in better performance for verification. While we have endeavored to make plausible assumptions with regard to our simulations, there is of course much scope for more fully mapping out the performance dependence on different techniques depending on individual gate errors and durations, and physical layout. The latter, in particular will likely make a great difference to the relative performance of the parallel ancilla technique. For example, the superiority of series based verification in our case was a direct result of ancilla “movement” (via SWAP operations) being no faster than ancilla recreation.

A model with faster ancilla movement or longer ancilla encoding time could easily change this. Our emphasis is on the fact that decoding shows demonstrably better performance in a range of cases, even when measurement is not slow, and that this advantage becomes apparent when one considers dealing with the practical consequences of verification failure: having to either recreate or move into position additional ancillas. Moreover we note that the additional errors induced by the latter can make a significant difference to overall error rates, and should be considered when assessing the performance of QEC techniques, including (as our work does not currently do) the consequences in a many-stage computation of, on occasion, occasionally failing to verify any ancillas, and the optimal choice of techniques in this scenario. In chapter 4, we study the possibility and the effect of applying fault-tolerant quantum error corrections FTQECs less often after a set of logical gates with the assumption that some of the quantum error corrections have been canceled due to the ancilla postselection process failures. The reason for that is, when fault-tolerant quantum error correction is implemented using noisy gates, the conventional tendency to apply quantum error correction after every operation is not always optimal for reducing overall statistical error. In order to determine how often quantum error correction should be applied to minimize the logical error rate, we present a mathematical relationship between the logical error rate and the number of logical gates after which the quantum error correction is applied using the [[7,1,3]] Steane code. Furthermore, we compare the numerical values of the logical error rate from the mathematical formula with these from the Monte Carlo simulation. These relationships reveal that logical error rate is minimized when QEC is performed after blocks of five gates. Our analysis demonstrated that QEC is actually necessary only after five logical gates (i.e.m = 5), since this makes the logical error rate P_L minimum. In addition, simulated implementations “noisy“ QEC demonstrate that

for a sequence of a logical gates it appears useful to apply QEC after every five logical gates.

BIBLIOGRAPHY

[1] D. P. DiVincenzo and P.Aliferis, “Effective fault-tolerant quantum computation with slow measurements,” Phys. Rev. Lett., vol. 98, p. 020501, 2007.

[2] M. Nielsen and I. Chuang, Quantum computation and quantum information. Cambridge University Press, New York, 2010.

[3] F. Gaitan, Quantum error correction and fault tolerant quantum computing. CRC Press, 2008.

[4] J. Preskill, H.-K. Lo, S. Popescu, and T. P. Spolle, Introduction to Quantum Computation. World Scientific, Singapore, 1998.

[5] N. D. Mermin, Quantum Computer Science: An Introduction. Cambridge University Press, 2007.

[6] M. Byrd, “Qunet.” http://qunet.physics.siu.edu/wiki, 2010.

[7] P. Brooks, Quantum Error Correction with Biased Noise. PhD thesis, California Institute of Technology, Pasadena, CA, 2013. http://thesis.library.caltech.edu/7774.

[8] D. Bacon, The Quantum Circuit Model and Universal Quantum Computation . University of Washington, 1995.

[9] A. Y. Kitaev, A. H. Shen, and M. N. Vyalyi, “Classical and Quantum Computation,” American Mathematical Society, vol. 47, 2002.

[10] D. Gottesman, “An introduction to quantum error correction and fault-tolerant quantum computation,” 2009.

[11] E. Knill, R. Laflamme, A. Ashikhmin, H. Barnum, L. Viola, and W. Zurek, “Introduction to quantum error correction,” 2002.

[12] W. K. Wootters and W. H. Zurek, “A single quantum cannot be cloned,” letters to nature, vol. 299, pp. 802–803, 1982.

[13] P. Shor, “Scheme for reducing decoherence in quantum computer memory,” Phys. Rev. A, vol. 52, p. 2493, 1997.

[14] A. Steane, “Error correcting codes in quantum theory,” Phys. Rev. Lett., vol. 77, p. 793, 1996.

[15] D. Trieu, Large-Scale Simulations of Error-Prone Quantum Computation Devices. and Distributor: Forschungszentrum Jülich GmbH Zentralbibliothek, Verlag D-52425 Jülich, 2009.

[16] A. M. Steane, “Overhead and noise threshold of fault-tolerant quantum error correction,” Phys. Rev. A, vol. 68, p. 042322, 2003.

[17] J. Preskill, “Reliable Quantum Computers,” Proc. Roy. Soc. London A, vol. 454, pp. 385–410, 1998.

[18] F. MacWilliams and N. Sloane, The Theory of Error-Correcting Codes. North- Holland Mathematical Library, 1977.

[19] N. I. sailovic, M. Whitney, Y. Patel, and J. Kubiatowicz, Proceedings of the 35th Annual International Symposium on Computer Architecture. IEEE Computer Society, 2008.

[20] A. Calderbank and P. Shor, “Good quantum error- correcting codes exist,” Phys. Rev. A, vol. 54, 1996.

[21] A. M. Steane, “Active stabilization, quantum computation, and quantum state synthesis,” Phys. Rev. Lett., vol. 78, p. 2252, 1997.

[22] D. Gottesman, Stabilizer Codes and Quantum Error Correction. PhD thesis, California Institute of Technology, 1997.

[23] A. Cross, “QASM.” http://www.media.mit.edu/quanta/quantaweb/ projects/qasm-tools, 2006.

[24] A. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, “Surface codes: Towards

practical large-scale quantum computation," Phys. Rev. A, vol. 86, p. 032324, 2012.

[25] A. Calderbank and P. Shor, "Good quantum error-correcting codes exist," Phys. Rev. A, vol. 54, p. 1098, 1996.

[26] Y. Weinstein, "Quantum error correction implementation after multiple gatest," Phys. Rev. A, vol. 88, p. 012325, 2013.

[27] Y. Weinstein, "Quantum error correction during 50 gates." http://arxiv.org/abs/1312.1245, 2013.

[28] A. Abu-Nada, B. Fortescue, and M. Byrd, "Relative performance of ancilla verification and decoding in the [[7,1,3]] steane code," Phys. Rev. A, vol. 89, p. 062304, 2014.

APPENDICES

APPENDIX A

DETAILS OF ANCILLA DECODING

In this section we describe the possible ancilla errors due to a single fault in the encoding circuit, and the corresponding error syndromes when using the ancilla decoding technique. This is an expansion of the summary given in [1]. We will consider the case of X errors due to faults in the encoding circuit for $|0\rangle_L$; the case of Z errors when encoding $|+\rangle_L$ is directly analogous.

Figure A.1 illustrates the circuit for encoding $|0\rangle_L$, which possesses two types of gates: (a) single qubit gates defined as preparation of $|0\rangle_L$and $+0\rangle_L$with waiting gates represented by bold line segments in Figure A.1, as well as (b) two qubit gates defined as CNOT gates. The decoding circuit can uniquely identify any single-qubit error. Two qubit-errors caused by a single fault will either be errors on both outputs of a CNOT gate, or a single error which at some point propagates to two errors via CNOT, thus it suffices to further consider only the case of a CNOT gate on which both outputs have X errors. These are listed in Table A.1 according to the encoding circuit labeling given in Figure A.1.

As shown, dual output X errors propagate to single X errors to the data in the cases of CNOT gates 4, or 5, or 6; or as two X errors to the data in the cases of CNOT gates 7, or 8, or 9. See Table A.1. However, for CNOT gates 1, or 2, or 3, dual X errors in the two output channels will propagate as four X errors on the data. These are equivalent to the X-stabilizer generators of the Steane Code, which accordingly do not affect the data. The stabilizer generators of the Steane Code are listed in A.2. Finally, for CNOT gates 4, or 5, or 6; the single faults which are produced in the two output channels will propagate as a dual X errors on the data.

These X errors (single and double X errors) on the data can be identified by the decoding circuit and so corrected. As seen, every multi-qubit error, is, up to stabilizers, equivalent either

to a single-qubit error or to one of three two-qubit errors,

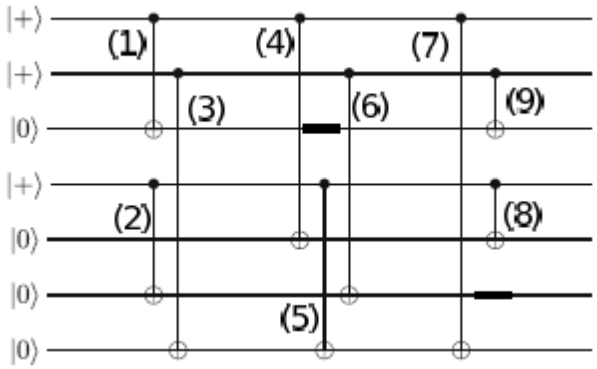


Figure A.1: Encoding circuit for $|0L\rangle$

CNOT	Multi-qubit data errors
1	$X_1X_3X_5X_7 \equiv G_1$
2	$X_4X_5X_6X_7 \equiv G_2$
3	$X_2X_3X_6X_7 \equiv G_3$
4	$X_1X_5X_7 \equiv G_1X_3$
5	$X_4X_5X_7 \equiv G_2X_6$
6	$X_2X_3X_6 \equiv G_3X_7$
7	$X_2X_3 \equiv G_3X_6X_7$
8	$X_4X_5 \equiv G_2X_6X_7$
9	$X_1X_7 \equiv G_1X_3X_5$

Table A.1: Examples of output errors produced by X errors on both outputs of CNOT

X-stabilizer	Z-stabilizer
$G_1 = X_1X_3X_5X_7$	$G_1 = Z_1Z_3Z_5Z_7$
$G_2 = X_4X_5X_6X_7$	$G_2 = Z_4Z_5Z_6Z_7$
$G_3 = X_2X_3X_6X_7$	$G_3 = Z_2Z_3Z_6Z_7$

Table A.2: The stabilizer generators of the Steane code.

VITA

Graduate School

Southern Illinois University

Ali Abu-Nada

email address: ali_nada@yahoo.com

Applied Science University, Amman-Jordan

B.S. Applied Physics 2001.

University of Jordan, Amman-Jordan.

M.S. Physics 2003.

Southern Illinois University Carbondale

M.S. Physics 2011.

Dissertation Title: The effect of the Ancilla Verification on the Quantum Error Correction

Major Professor: Mark Byrd

Publications:

Gary Leuty, Ali Abu-Nada, and Mesfin Tsige. “Multilayer Adsorption of Methane and Chloromethane on theMolybdenum (100) Surface”. J. Phys. Chem. C 116, 14514 (2012).

Ali Abu-Nada, Ben Fortescue, and Mark Byrd. “Relative performance of ancilla verification and decoding in the [[7, 1, 3]] Steane code”. Phys. Rev. A 89, 062304 (2014).